Project One, CS444, Spring 2018, Homework Group 10

Kamron Ebrahimi, Maximillian Schmidt & Brendan Byers

ebrahimk, schmidtm, byersbr

April 15, 2018

Abstract

This report details the process of building the Linux Kernel and running the kernel on the Oregon State University, operating systems II server. To accomplish this task, the command line based, virtual machine, QEMU, was cloned and configured in order to produce a safe, experimental environment in which the Linux Kernel could be modified. This paper will walk the reader through the key components of configuring this environment. The team was able to successfully build the Kernel and boot the Kernel within the QEMU virtual machine.

CONTENTS

I Commands Used 2

II QEMU Command Explanation 3

III Git Log 3

IV Work Log 4

V Scripts 5

I. Commands Used

In order to run the Linux Kernel in a QEMU virtual machine a folder titled ”10” was created with the following file path, /scratch/spring2018/10. A copy of the linux-yocto repository was then cloned and unzipped to the group 10 folder using the following commands from within the group 10 folder.

- **wget http://git.yoctoproject.org/cgit.cgi/linux-yocto/snapshot/linux-yocto-3.19.2.zip**
- **unzip -q linux-yocto-3.19.2.zip**

With the linux-yocto repository successfully cloned the QEMU environment needed to be set up and run. Some preliminary measures were required to ensure this was performed properly. First a script located at the file path /scratch/opt/environment-setup-i586-poky-linux was sourced. All members of group ten operate in the bash shell, thus this script at the aforementioned file path was used. To source this script the following command was executed from the root directory of os2.engr.oregonstate.edu.

- **source /scratch/opt/environment-setup-i586-poky-linux**

In order to ensure that the Makefile associated with the Linux Kernel executed properly, the file located at the file path /scratch/files/config-3.19.2-yocto-standard was copied to a newly created folder named ”.config” located within the linux-yocto-3.19.2 folder. Additionally the starting Kernel image, bzImage-qemux86.bin and drive file core-image-lsb-sdk-qemux86.ext4 were copied within the linux-yocto-3.19.2 folder. To accomplish this the following commands were executed from within the linux-yocto-3.19.2 folder.

- **cp /scratch/files/config-3.19.2-yocto-standard ./.config**
- **cp /scratch/files/bzImage-qemux86.bin .**
- **cp /scratch/files/core-image-lsb-sdk-qemux86.ext4 .**

With the aforementioned preliminary configurations complete, the clone of the Linux Kernel could be compiled. The following command allocates four threads to build the Linux Kernel. A limit of the number of threads is used to ensure that no one group consumed too many computational resources on the shared os2.engr.oregonstate.edu server.

- **make -j4 all**

With the Kernel built, the last step was to boot the Kernel from within an instance of the QEMU virtual machine. First an instance of QEMU was executed on port 5510 of the os2 server using the following command:

- **qemu-system-i386 -gdb tcp::5510 -S -nographic -kernel bzImage-qemux86.bin -drive file=core-image-lsb-sdk-qemux86.ext4,if=virtio -enable-kvm -net none -usb -localtime –no-reboot –append ”root=/dev/vda rw console=ttyS0 debug”**

Using GDB from another terminal window the Kernel could be executed. To complete this a group member had to connect via port 5510 via GDB to the QEMU virtual machine and execute the ”vmlinux” process from within the linux-yocto-3.19.2 folder. The following chain of commands were used while an instance QEMU was running on port 5510, these commands were executed from within the group 10 folder.

- **$GDB linux-yocto-3.14/vmlinux**
- **target remote :5510**
- **continue**

The above commands create a gdb connection to port 5510 of the os2 server and enters the vmlinux file. The vmlinux executable when run hoists an instance of the Linux Kernel from within the QEMU virtual machine. By running the ”continue” command from within the gdb connection, vmlinux is executed which boots the Kernel from within QEMU.

II. QEMU Command Explanation

- **-gdb tcp::5510**
  - This will cause QEMU to wait for a connection on a device, which in this case is a tcp port. The connection can also be UDP, pseudo TTY, or standard output. Using standard output allows you to start QEMU from within gdb and establish the connection with a pipe.
- **-S**
  - This signals to qemu to not start the CPU at startup.
- **-nographic**
  - This will start qemu with graphical output completely disabled. The emulated serial port will usually be redirected to the console.
- **-kernel bzImage-qemux86.bin**
  - Points to the kernel image. This image can either be a standard linux kernel or a kernel in multiboot format.
- **-drive file=core-image-lsb-sdk-qemux86.ext4,if=virtio -enable-kvm**
  - The -drive defines a new drive, which includes creating the block driver and the guest device. Our drive in this case is . Next, it checks if virtio is enabled and enables the kvm accordingly. Virtio is a virtualization standard that allows the guest to know that is is in a virtual environment.
- **-net none**
  - This is used to create an onboard Network Interface Card. In our case we dont want to create any network interfaces for our kernel. If this option isnt specified, then a single NIC is created.
- **-usb**
  - This flag will enable the usb driver for the guest vm.
- **-localtime**
  - This is used to specify that the vm should use local time. This flag has been depreciated as of qemu 2.12.0 and was replaced by the -rtc flag.
- **–no-reboot**
  - This will stop qemu from exiting when the guest shuts down. Instead, it will only stop the emulation.
- **–append ”root=/dev/vda rw console=ttyS0 debug”**
  - This will tell the emulator to use the command line in quotes to be used as the kernel command line. This line here dictates the location of the root directory, what sort of console, and to boot in debug mode.

III. Git Log

Date	Author	Description
9/4/2018	Maximillian Schmidt	Copied yocto VM files
9/4/2018	Maximillian Schmidt	Created .config folder and moved config-3.19.2-yocto-standard file into .config
10/4/2018	Maximillian Schmidt	Copied bzImage-qemux86.bin and core-image-lsb-sdk-qemux86.ext4 into yocto environment

IV. **Work Log**

To track assignment submissions and homework workflow, a git repository was created. All tex and make files are organized within this repository. Below is a detailed repositroy log for project number one. The vast majority of work setting up the yocto-QEMU enviroment was performed at the beginning of week two. After the environment was configured and the Kernel was booted within the QEMU virtual machine each team member was briefed on the configuration process and work on the tex document was started.

Link	**Date**	**Author**	**Description**
e10cf2a	Tue Apr 10 22:42:32 2018 -0700	ebrahimk	added Makefile and tex template
399064e	Tue Apr 10 22:45:33 2018 -0700	ebrahimk	made project1 folder
7e41f8e	Tue Apr 10 22:51:18 2018 -0700	ebrahimk	Added another tex template
90bed41	Wed Apr 11 13:41:22 2018 -0700	cascadeth	Added scripts for easy setup of the VM for hw1
225209c	Wed Apr 11 17:35:33 2018 -0700	ebrahimk	Progress on tex report
a22de85	Wed Apr 11 17:43:53 2018 -0700	ebrahimk	IEEEtran styling added
466bf6d	Wed Apr 11 20:22:08 2018 -0700	ebrahimk	Git log added for reports
bfdf3b3	Wed Apr 11 20:34:19 2018 -0700	ebrahimk	remove log
604b920	Thu Apr 12 15:22:05 2018 -0700	cascadeth	Updated script with colors and clarifications
aed5444	Thu Apr 12 16:24:00 2018 -0700	cascadeth	updated gitignore
c18877d	Thu Apr 12 16:32:36 2018 -0700	cascadeth	removed .DS-Store files; now included in gitignore

V. Scripts

Below is an environment configuration script used to automate all of the required setup for project one.

```
#!/bin/bash

GROUP_NUM=10

PROJECT_DIR="/scratch/spring2018/$GROUP_NUM"

# The following commands setup the necessary environment for HW1

# Make the group's directory (group 10) if it doesn't exist
if [ ! -d "$PROJECT_DIR" ]
then
        echo "Making new project directory..."
        mkdir "$PROJECT_DIR"
fi
echo "Project directory exists - moving to.."
cd "$PROJECT_DIR"

# Download the proper kernel
echo "Getting yocto kernel..."
wget http://git.yoctoproject.org/cgit.cgi/linux-yocto/snapshot/linux-yocto-3.19.2.zip

# Unzip it here (no stdout)
echo "Unzipping in current location..."
unzip -q linux-yocto-3.19.2.zip

# Move into the new dir
echo "Moving into new dir..."
cd ./linux-yocto-3.19.2

# Copy the proper config file
echo "Copying source config files..."
cp /scratch/files/config-3.19.2-yocto-standard ./.config

# Copy the proper initial binary and ext4 file system
cp /scratch/files/bzImage-qemux86.bin .
cp /scratch/files/core-image-lsb-sdk-qemux86.ext4 .

# Begin compiling the new kernel
echo "Compiling new kernel for VM..."
make -j4 all
```